Book of Kali:
Privacy Fundamentals

Author: Seccult
Date: April 13’th 2026
Version: V1.0

Table of Contents:

Introduction: - pages 1-2

Tor browser setup, root/sudo, + adduser command: - pages 3-5

Welcome to “Cyberia”! Tor Browser Resources, and Websites: pages 6-8

Configuring Proxy Chains: - pages 9-12

Using Proxy Chains With Other Applications: - page 12

Running Proxy Chains With Tor: - pages 12-13

Finding Proxies Online Information and Tips: - page 14

Virtual Private Network Setup/Tutorial: - pages 15-19

How To Disable webrtc: - page 20

Changing Ones MAC Address With “maccganger” : - page 20-22

“crontab” Tutorial + “macchanger” Startup Script: pages 23-26

Other OPSEC tools: pages 26-28

Conclusion: page 28

Introduction:

whoami: So here is the deal, I am a multi certified pentester, and defense analyst, for years I've survived as a scrappy freelancer in the industry.

I very recently took a job offer that spoke to my heart, but prior to this position I turned down several six figure job offers because they sounded fucking retarded, and I think I woulda kissed the end of a shotgun had I taken them.

I'm a high school dropout, never went to no fancy university, my set of knowledge came from being self taught, and weaponizing my autism.

When I started my journey, it came out of a place of love for privacy and freedom, I wanted to understand how these systems worked, and that was enough to carry me through.
These guides/tutorials have been painstakingly converted from my original handwritten notes, and as I complete converting more I will upload them here free of charge.

I'm doing this because when I was starting out I had a difficult time getting help, and I got put down so much, some IRL people I knew actively discouraged me from even attempting to learn. Now I am in a position where I'm living my life doing the thing I love, for reasons I feel good about, and I reached something people spend their entire lives seeking, fulfilment.

If I listened to everyone telling me I couldn't, or shouldn't do something, I wouldn't be here, I said fuck em instead, and I did the thing a lot of people thought I couldn't do, it wasn't always easy, but I did it. I'm writing all this out to give some context, and to say you can learn about these systems from zero, you don't need to go to university, or pay exorbitant amounts to do so, and if you genuinely want to learn, these are my notes free of charge to help you.

Book of Kali Privacy Fundamentals Overview:

This second manual is meant for people starting their journey from zero practical linux experience, it's not a theortical course, and builds directly off of the knowledge from my first manual **The Book Of Kali: Basics.**

That said, this manual puts an emphasis on basic privacy considerations when using Kali Linux, and helps one familiarize themselves with some of the built in tools, and commands that can be used for OPSEC purposes. Beyond these subjects, it also delves deeper into common linux system usage, and touches on scripting.

Unlike my previous manual which was meant specifically for beginners, this manual may also be of interest for those who already hold linux fundamentals, and understand how to operate a terminal, but want some insight on some OPSEC TTP's when using Kali Linux.

The next courses after this one will go deep into a specific subset of offensive security, but this manual is going to be a pre-requisite to that course, as some of the tools used in the next manual will

be discussed in this manual first, so if you are a beginner to either offensive, or defensive security you must read the manuals in order.

Note: This tutorial was written with Kali in mind, you can use another OS, and it should be okay, but you may have issues following certain sections if you do.

Say you've managed to do that? Well welcome, doing these courses (in order) will take you from linux virgin riding the internet bareback, to someone proficient enough to use protection, ready to do a bit of penetration testing .

This guide is the second in a set of guides meant to teach one the fundamentals, if you do them all, you are going to get a foundation for linux usage, and defensive tools, before I release my next course on some offensive tools (again free of charge).

You won't become a 1337 hacker, but you will gain enough knowledge to continue learning, and operate a terminal comfortably, and move laterally from there.

There are going to be 2 guides in this series:

1). Basics.
2). Foundational Privacy.

One more thing:

Please feel free to offer some suggestions, but just be aware I'm releasing these guides for free, and I may not be able to make changes right away (or at all), sorry, I’m fucking crazy busy.

Note about copywrite: This document is free to read, download, share, just link to the official source, and please give me credit.

Tor browser setup, root/sudo + adduser command:

Background: The **Tor browser** anonymises the user while searching the web. The Tor browser also allows the user to access the **"dark web"**, or the **"deep web"**, also known as the **"hidden"**, or **"secret"** web.

The "dark web", can only be accessed with Tor, and is a phenomenal tool against censorship, and a great resource to find tutorials; and further insight into pentesting. Additionally there are also hidden marketplaces where one can buy goods with cryptocurrencies.

These hidden websites use a **.onion** domain suffix, used to designate them as a hidden service, A list of these .onion websites will be provided as a reference in a later chapter.

Note: The Tor browser setup requires some pre-requisite knowledge of **sudo,** as well as the **adduser** command in order to set it up properly, as well as to use it securely later. For these reasons both **sudo**, and the **adduser** command will be discussed first prior to seting up, and operating the tor browser.

"root/sudo" Overview, and usage:

Overview: A "superuser" is an individual on a linux system that has **root privileges**, meaning that they are able to access, and use **ANY** of the system resources, tools, documents, ect. This superuser is known as **sudo** in shorthand on these systems, and a **root** user is ostensibly the same as a **sudo** user.

root (or sudo) would be the linux equivalent to a **system** user on windows systems, and has also been referred to as "**god**" mode by some hackers, due to it's comparable nature to a mode of the same name present in some videogames.

Older versions of Kali Linux had the user set to **root** by default, the rationale at the time was that Kali linux was an "advanced" level operating system designed to be used by those already familiar with linux systems, and the fundamentals of user security. So by enabling the **root** user by default, it would save time when running offensive tools that required **sudo** permissions.

Current versions of Kali linux no longer set the user as **root** by default, instead in order for one to operate as **root**, or a superuser with **root** permissions, one needs to switch to these users, by using the **sudo su** command.

A word to the wise, on"sudo", and "root" users:

Because these users operate using elevated permissions, if an attacker were to gain access to a system running as either of these users, or alternatively managed to exploit a vulnerability that allowed a user to become root, the system would be considered completely compromised.

As such, any actions one does as these users such as downloading, executing, and running programs must be done with extreme caution, due to the inherent permissions these accounts have, and

potential risk if a malicious program were to be executed with root permissions.

How to switch to a root, or superuser account:

To switch to a root user one can use the following command:

sudo su root

After running this command one will be promted for the root user password, this is the password of the “main” account of the system.

Note: After successfully executing this command, the user name within the terminal will change, and be reflected in the following format **root@kali**

To switch to another user with super user privileges one can use the following command:

sudo su [user name]

After running this command one will be prompted for the sudo user password, this is the password of the account with elevated root permissions on the system.

Note: After successfully executing this command, the user name within the terminal will change, and be reflected in the following format **[sudo username]@kali**

The “adduser’ command:

The **adduser** command will also be covered as part of the Tor browser setup, this is because one will need to create a seperate user for when they utilize Tor.

WARNING: ONE SHOULD NOT USE TOR AS A ROOT USER, DOING SO WILL COMPROMISE ONE’S SECURITY!

adduser This command will add a new user, after doing so one may also configure a new password for this user.

Note: The UNIX password one types in will be invisible as one types it, this is normal, and is done to protect an individuals privacy.

Command Format Example:

adduser [username]

Practical Command Format Example:

adduser mrrobot

The above command will create a new user **mrrobot.**

Note: It is important to note that usernames **CAN NOT** contain capitals!

Lastly, one will be prompted for some additional information for the newly created user, full name, room number, ect... This information can be inputted, or left blank if you so choose.

Once one is satisfied, they may press the **y** key to confirm the new user account creation.

Note: For reference the **adduser** command is different than the **useradd** command.

Tor browser setup instructions:

1). As a **root** user, one must type in the following command:

sudo apt update

Followed by:

sudo apt install -y tor torbrowser-launcher

The first command will first update your kali, the second command will the install tor on your Kali instance (including the signature verification), as long as one has a stable internet connection.

2). Once successfully installed one must **change** to a **low privileged user** prior to launching the tor browser, to do so, within the **root terminal session** one must use the following command:

exit

This command will **terminate** the root user terminal session, and revert one to the prior user account.

Note: Attempting to launch the Tor Browser as a **root** user will fail, as it is forbidden to do so under new iterations of the tool, this is a preventative security measure.

3). One may now switch to a low privilegd user, or alternatively execute the tor browser utilizing the current user by issuing the following command:

torbrowser-launcher

4). If successful, the tor browser will now launch, and be available for use for the current user.

Fun fact: I needed to re-write this entire section, as back when these notes were written getting Tor on one's Kali linux OS used to be so much more difficult, you young bloods don't know how good you have it.

Welcome To Cyberia! Tor Browser Resources, and Websites:

Resources: Using the following search terms should hopefully help one find the resources they may need on the “dark web”, a brief description of the resource is also included. Tor websites often change their address, the following entries in this section are supposed to be used as “search terms”, for future reference:

Hidden Wiki: The “hidden wiki” is a listing of significant **.onion** domains. Some of these links are for illegal services, use discretion when visiting links on the “Hidden Wiki” directory page.

Dread Forums: A reddit sytle deep web forum that discusses both illicit, and non illicit topics, often used by “carders”, “privacy enthusiasts”, “hackers”, and “dark net marketplace” vendors, and consumers.

Secure Drop: A website designed for whistle blowers to upload sensitive documents anonymously.

Deep Web Glossary Overview:

I debated on whether to include this section since some of these terms are offensive, or pretain to illegal material. Ultimately, I felt it's import to include them so those utilizing the deep web understand the slang, and can thereby avoid illicit material, or to proceed with caution if visiting hacker forums for OSINT purposes. **This list is not definitive!**

Blackhat: A criminal hacker that is willing to break the law for profit irregardless of the damage caused.

Grey Hat: A hacker that is willing to break the law, but is not necessarily malicious, this term normally applies to activist hackers who break the law.

Whitehat: A hacker that works in the field legally, usually refers to government workers.

Greenhat: A freelance hacker, or mercenary for hire, refers to a hacker that hacks for money based on bug bounties or contracts, can be legal, or illegal.

OSINT: Stands for open source intelligence, or information that can be found freely.

Carder: A deep web vendor who sells stolen credit cards, or the tools to steal credit cards.

RAT: This stands for remote activated trojan (sometimes referred to as a remote administration tool), a often malicious tool that allows the attacker to access the victims computer remotely.

Social Engineering: The discipline of using social interaction, manipulation, and chicanery to get the victim to perform an action they otherwise would not. This kind of hacking relies less on technical computing skills, and more on social manipulation.

Gibson: Hacker slang for a computer.

Box: Slang for a computer .

Glowie: Hacker slang for a federal agent, also sometimes referred to as a “glow nigger”, this was a slang term coined by schizophrenic programmer Terry Davis, and adopted by the underground hacker community.

Alphabet boy: Usually refers to a government agent of the FBI, CIA, NSA ect.

Anon: Someone who is anonymous online.

Vanned: Refers to a hacker that was arrested.

Moral Fag: Hacker slang for a hacker that hacks for moral reasons.

Name Fag: Hacker slang for a hacker that hacks under their own real name.

Ego Fag: Hacker slang for a hacker that hacks to prove a point about their abilities.

Leader Fag: Hacker slang for someone who runs a hacking outfit.

CODC: The cult of dead cow, a legendary, and highly secretive hacking group, responsible for lot’s of 0-day exploits.

GNAA: The Gay Nigger Association of America, another highly respected hacking collective, that mostly hacked for fun/hacktivist reasons.

Anonymous: A activist hacking collective.

1337: Hacker slang for leet, or elite, refers to someone, or something that is very good.

Hard Candy: Refers to child porn.

Popping a box: Getting control of a computer.

0-day: A kind of exploit that is newly discovered, for which no patches exist because it is so new, these kind of exploits can be sold for millions.

Creds: Stands for user credentials, usually these are sold online through carding websites.

Spoofing: Using faked information.

Phishing: A kind of generalized social engineering attack, where an email is sent to the victim in the hope they will click a malicious link, or download malicious software.

Spear-phishing: A kind of highly specific social engineering attack against a single individual, where a highly personalized email is sent to the victim in the hope they will click a malicious link, or download malicious software.

Vishing: A kind of social engineering attack, where a phone call is placed to the victim in the hope they will perform an action they otherwise would not.

Crypto: Refers to online currencies. Such as bitcoin, monero, ect.

Pwned: When someone, or something has been owned, usually refers to a hacker gaining access to a computer system.

Script Kiddie: Derogatory term that refers to new, or bad hackers that just run scripts without understanding what they do, or how they function.

N00b: A derogatory term to refer to someone who is new to the internet, gaming, hacking, ect.

Doxxed: Refers to someone whos real life personal information has been disclosed, usually in a involuntary manner.

RTFM: This acronym stands for “Read The Fucking Manual”, and is usually said to someone who asks basic information on tool usage, that can be found easily by using the **man**, or **help** command options.

Waifu: Can refer to a girlfriend, wife, or female in general, both real, or fictional, derives from anime.

Warez: Pirated software distributed online usually via a torrent site such as the pirate bay, or the dark web.

Sailing the high seas: A term used to describe using a torrent site to seek, and download content, using associated with the website the pirate bay.

Nerdcore: A type of music often associated with rap songs that deal with computing, especially black hat subjects, popular artists in this field include YTcracker, Dual Core, Mc Frontalot, with arguably the most popular nerdcore artist to become mainstream being K-Flay.

Try Harder: Offensive Security's (now Offsec) infamous old motto. Offsec is the creator of Kali Linux, and the administrator of the OSCP course, and exam, which is considered the gold standard for certifications in the penetration testing field. The motto is generally used by prior students, but has become associated with the “hacker mindset”, and is often used towards newcomers to infosec, often in a derogative manner. It's popularity is evident in that the motto was included in the lyrics of the Ytcracker song “Thrown I Own”, and a rap song “Try Harder” by Uzimon.

The Dark Owl Glossary: This resource provides a more comprehensive glossary of slang, and terms associated with the deep web, and computing in general:

https://www.darkowl.com/resources/darkowl-glossary-of-darknet-terms/

Configuring Proxy Chains:

Overview: Proxy chains give the operator the ability to route their traffic through a series of proxy servers, and stay anonymous by "hiding" behind them.

Or by having them forward ones request; so that it appears to the observer on the other side that the requests are being emitted from the proxies, instead of directly from the operator.

Note on free proxies: While there are a lot of free proxy servers available for use, they are not very stable, as such they are not useful for brute force attacks, or mass scannings.

Proxy chain configuration instructions:

1). In the terminal type in the following command:

nano /etc/proxychains.conf

The above command will open the proxy chains configuration file, and allow one to set up the proxy chains to their liking.

2). There are 3 different kinds of proxies one may use:

- **HTTP**
- **SOCKS4**
- **SOCKS5**

Proxies one finds online can be added to the configuration file, by including them under the section labelled **[ProxyList]**.

Note: More information on how one can add proxies will be covered in the chapter **adding proxy chains to [ProxyList] instructions**.

Note II: One always wants to make sure to use **SOCKS5** proxies, as these proxies are the most secure, as **SOCKS4** does not support **IPV6**, or the **UDP** protocol.

3). The following are setup instructions for different kinds of proxy chains, and a description of what they do.

Proxy chain types setup instructions:

In the **/etc/proxychains.conf** file there will be several different kinds of proxy chain types listed.

In order to **enable** a specific kind of chain to run, one must delete the **hashtag symbol #** next to the proxy option they wish to run, refer to the following examples below:

Disabled dynamic chain code example:

#dynamic_chain

Enabled dynamic chain code example:

dynamic_chain

To reiterate, If one were to want the **#dynamic_chain** to run, they would **delete** the **#** symbol next to it, afterwards one should save the file to apply the setting. So the above example would then read **dynamic_chain** **WITHOUT THE HASHTAG SYMBOL,** this will enable the option.

Note: All other proxy chain options that one is not going to use should have the **hashtag symbol #** next to them, so they are disabled, otherwise this will cause errors if incompatible chains are enabled at the same time.

Proxy Chains Option Descriptions:

dynamic_chain Each connection will be done via chained proxies, all the proxies will be chained in the order they appear in the list, if an inactive (dead) proxy is in the list, it will be skipped, and the chain will use the next available proxy (at least one proxy must be online to work in the chain).

Otherwise EINTR is returned as output.

Note: Generally a dynamic chain is the kind of proxy chain one should use.

strict_chain Each connection will be done via chained proxies, all proxies are chained in the order they appear in the list.

Note: strict chain proxies **must go through the order the appear on the list, they will not work otherwise**! If a proxy on the list is down it will not work.

random_chain Each connection will be done via a random proxy (or proxy chain, see **chain_len**) from the list. This option is good to test ones IDS :}

Note: random chains are a way of “resetting” ones services. For example with **random_chain** one can specify a list of IP’s, and tell their computer when connecting to different points to use specific proxies a set amount of times before switching to another proxy chain.

Random chains are a way of creating custom chains for specific purposes, specific options can be used with **chain_len** (this stands for chain length).

quiet_mode No output is given from library.

proxy_dns Proxy DNS requests will make sure there is no leak for DNS data.

Note: This option is **EXTREMELY IMPORTANT** to have enabled! This option prevents DNS leaks, DNS leaks is a way one can be identified.

This is because, even though one can conceal their IP address with a proxy chain, one can still be identified if there is a DNS leak. The reason for this is because an adversary can get the IP address of the DNS server, and this in turn can be used to figure out ones physical address, at which point they will no longer be anonymous.

A DNS server resolves domains to IP addresses, and vice versa. For instance, if one were to type in **youtube.com**, the DNS server of ones local ISP provider will resolve that into an IP address YouTube has, abd it will make a request, in doing so ones local DNS will be discovered.

4). The following will be an explanation on how to setup, and add proxy chains using proxies one finds online to the **[ProxyList].**

[ProxyList] Instructions:

The following is a detailed overview on the proxy format, as well as an explanation on this format.

Format For Entering Proxies:

(a)	(b)	(c)	(d)	(e)
socks5	192.168.67.78	1080	lamer	secret

The above is an example of a proxy chain format, that means how a proxy chain should be added to the **[ProxyList]** chain. A further explanation for each section of the proxy chain format is found below, and is correlated to the letters above each section.

Proxy chain format explaination:

a). The type of proxy being used, **http**, **socks4**, **socks5** (one should only use **socks5**).

b). IP address of the proxy server.

c). This is the port number on which the proxy server is listening, that is to say the port that is “open”.

d). Proxy server user name, not all proxies have user names, but some paid ones do.

e). Proxy server password, not all proxies have passwords, but some paid ones do.

Tab key This key can be used to neatly separate the different sections of a proxy chain format, that one may add.

SOCKS5 Proxy That Must Be Added To Chain:

socks5 127.0.0.1 9050 The following proxy is one that must be added to ones proxy chain list, the IP

address on this proxy (**127.0.0.1**) is the **loopback address** of ones local computer, it is used for interdevice communication, pinging this IP ensures the IP protocol is set up correctly.

Using Proxy Chains With Other Applications:

Once configured one may use proxychains with other applications to route the application traffic through proxy chains with the settings set within the configuration file. To do so one needs to just append the **proxychains** command prior to the name of the application one intends to use with proxy chains.

For example if one wanted to use proxy chains with NMAP, one would type the following command into the terminal:

proxychains nmap

Note: One should be aware to include the **proxychains** command before any application, and any arguments that eould follow normally.

Using proxychains with another application command format example:

proxychains [application]

Running Proxy Chains With Tor:

Once one has configured their **proxychain configuration file** to their desired specifications, they may now run these proxy chains with the **TOR browser**, and verify they're working properly with **DNSleaktest.com**, using the following tutorial:

1). Check Tor status:

First off one should check the status of Tor to see if it is running, to do so one should use the following command:

service tor status

2). If Tor is not running, one may use the following command to start the tor service:

service tor start

3). Once Tor has been started, and is verified as running, one now needs to verify the proxies are working as intended via DNS leak test, to do so use the following command:

proxychains firefox www.duckduckgo.com

This command will run proxychains with Tor as you use the firefox application, to give one an extra layer of added **anonymity.**

Once one has been directed to the **duckduckgo** search engine, type in the following search term into the search field:

DNS leak test

This should being up DNSleaktest.com (or something similar), one can use this resource to verify their IP address has been changed.

Note: duckduckgo.com should be ones primary search engine, this is because duckduckgo.com claims to not track it's users, making it ideal for privacy. Unfortunately in recent times duckduckgo.com has suffered some critisism due to privacy, and censorship reasons, for this reason the **Brave browser** may also be an alternative to consider.

4). Once you are done using the Tor service, and you no longer desire to operate anonymously, one can stop the service using the following command:

service tor stop

Note: Running Tor will cause performance issues when using the web, for that reason one should turn it off if there isn't a need to be anonymous online.

VERY IMPORTANT NOTE: For all the outlined steps to work, all browser windows must **NOT** be currently running! One should ensure firefox, chrome, ect are not running!

Confirm the terminal output to verify proxychains are running, after executing the **proxychains** command.

Reminder: One should **NOT** use tor as a **root** or **sudo** user.

Troubleshooting Tor:

Sometimes when using tor, it may not function, even when performing all steps correctly. This may be because some IPs tor can assign one may be banned. If this occures, do not panic, simply restart tor to obtain a new IP until it works.

Resetting the Tor service:

If one needs a new IP, they can reset tor by using the following command:

service tor restart

Finding Proxies Online Information and Tips:

One can find, and import proxies online for free, to add to ones **[ProxyList]** section in the **proxychains.conf** file.

In a desired search engine one should input the following search term (or something similar) into the search field:

free sock5 proxies

Once one has found some proxies online, they will need to obtain the **proxy IP**, **port number**, and **version information** in order to import them into ones **[ProxyList]**.

Please refer to the prior **[ProxyList] Instructions** for more information:

Best Proxy Jurisdictions:

When one is selecting proxies to add to thier **[ProxyList]** it is important to keep in mind certain jurisdictions where the proxies are located will offer more privacy, and security than others.

The best jurisdictions are as follows:

1). Netherlands

2). Germany

Note: This list is outtdated, and I could use some suggestions to get it updated with relevant information.

Maximum Amount of Proxies One Should Use:

One should limit the amount of proxies they intend to add to the **[ProxyList]** chain to 2-3 maximum. Any more than that, and proxychains will become impractical to use as a result of performance issues.

Things to remember about proxies:

1). Not all proxies one finds online will work! One should check their proxies often to verify that they are still working, and if not update their **[ProxyList]** with new proxies.

2). When running proxy chains one should check the terminal output to understand how traffic is being processed through your configured proxy chains.

Virtual Private Network Setup/Tutorial:

Overview: A Virtual Private Network (or VPN) is technology that connects a private network across a public network to enable users on a private network to send, and receive data across shared, or public networks as if their computing devices were directly connected to the private network.

Applications running across a VPN may use the functionality, security, and management of a private network.

Setting Up A Virtual Private Network:

Prerequisite note: This entire process should be executes as **root**, or a **sudo** user, as it requires elevated privileges to modify configuration files.

1). In order to configure VPNs to work with your network manager, one must first make some changes to the configuration file, to do so one may use the following command:

nano /etc/NetworkManager/NetworkManager.conf

Once in the configuration file one will need to make a change to the status of the following line of code:

[ifupdown]
managed=false

The status of **managed=false** needs to be changed so it reads **true**, and the code reads as follows:

[ifupdown]
managed=true

Once changed one should use **ctrl+O** to save the file, then exit. In order for the configuration file changes to work one must restart the **network-manager**, to do so one may use the following command:

service network-manager restart

2). To protect ones privacy while using a VPN one must change the address of their DNS servers. Ones actual LAN address will forward their traffic to their ISP, who has a DNS server; and this in turn can be used to resolve ones queries, and identify them.

One may modify their DNS servers by using the following command:

nano /etc/dhcp/dhclient.conf

Once one has used this command to access the configuration file, they should proceed to scroll down

until they find the following line of code:

#prepend domain-name-servers 127.0.0.1;

Delete the # symbol so the line of code instead reads:

prepend domain-name-servers 127.0.0.1;

Once one has done this, they may then proceed to delete their current name server **127.0.0.1;**

Note: Although the vast majority of computers use the **127.0.0.1;** nameserver address, some other computers may use an alternative one; regardless one should delete the address!

Once this has been deleted one can now add in their new nameservers. The following nameservers that are provided below have been found from **www.opendns.com**, as well as **googles DNS:**

From www.opendns.com**:**

a). 208.67.222.222

b). 208.67.220.220

From Google DNS:

c). 8.8.8.8

To add all three to the configuration file, one can use the following command format:

prepend domain-name-servers 208.67.222.222, 208.67.220.220, 8.8.8.8;

Once one has changed their nameserver addresses to the new ones, they may the save the file using **Ctrl+O** then exit the file, and restart the **network-manager**, by inputting the following command into the terminal:

service network-manager restart

This should put ones saved changes into effect.

Note: 3 nameservers is the maximum amount one can input, doing so ensures ones actual nameserver never gets used, thereby protecting their privacy.

Verifying name server addresses have been applied:

If one wishes to check what nameserver addresses they are currently using, they can do so by using the following command:

cat /etc/resolv.conf

Lastly one can verify their new DNS servers are working correctly by using the following website:

https://www.dnsleaktest.com

Once on the DNS leak test website, select the **standard test** option. After doing so, if one is using the https://www.opendns.com nameserver addresses, under the ISP column it should read "**OpenDNS, LLC**".

Note: If one is unable to change the DNS server after they run the **cat /etc/resolv.conf** command, one should use the following commands to renew the "lease" and get new settings:

dhclient -r

dhclient -v

Troubleshooting: If one is unable to change their DNS settings using the above steps, they may need to do so manyally using the NetworkManager GUI application, or by adding the DNS addresses to the **/etc/resolv.conf** file directly. The commands needed to do this may also be outdated, and instead of **network-manager**, one may need to use **NetworkManager**, if I get enough feedback I may revise this section with more relevant information.

A note regarding DNS server selection:

While the following nameserver addresses were chosen from **www.opendns.com**, because that service is safe, and secure, one does not need to use that specific nameserver address if they don't wish to.

In certain situations it may make more sense to use a DNS from the same country as ones proxy, or VPN, so there are less discrepancies between the two, and therefore less suspicions raised.

For instance, using a VPN from Germany when ones DNS is from the United States, is an inconsistency which may cause suspicion to an observant analyst.

3). Once one has changed their DNS name servers, they may obtain VPNs from the VPN Book web resource at the following address:

https://www.vpnbook.com

Upon reaching the site, one should go to the **free VPN** section of the site.

There should be several different VPN packages one can download, one should then select the desired VPN they wish to download. Once one has chosen a VPN, they should download it, the selected packages will then be downloaded to the **downloads directory**.

From the **downloads directory** one should move the VPNS to a desired directory (ideally a directory named VPNs, with sub directories named after the country of each VPN package downloaded).

From the **https://www.vpnbook.com** website, be sure to remember to **write down** the **user name** and **password** found on the **free VPN** section page, as one will need these credentials to get the VPNs to work.

Note: It's very important to remember the password occasionally may change, so if a VPN isn't working, one should just be cognizant of the current working password, and use that one.

4). Once one has downloaded their desired VPNs, and written down the user name, and password, the next step is to close their browser, and any tabs open.

Note: This is very important, the VPNs wont work if the browser is running.

Once the browser, and any remaining tabs have been closed, one can run the VPN by using the **openvpn** command in the terminal, and targeting the path of the specific VPN, as one can observe in the examples below.

Running a VPN command format example:

openvpn [path to .ovpn file]

Running a VPN practical command format example:

openvpn /root/Desktop/VPNS/GermanyVPNS/vpnbook-de233-tcp443.ovpn

The above command will run the VPN **vpnbook-de233-tcp443.ovpn** located in the **GermanyVPN** sub-directory of the **VPNS** directory located in the **Desktop**.

Once the VPN is running, input the username, and password (from vpnbook.com) when prompted, and the VPN should work.

5). The last thing one should do is confirm the VPN is working properly by using the **DNS leak test** resource, to verify their IP has changed, to do so one may visit the following link:

https://www.dnsleaktest.com

How to stop openvpn:

Once one is done using the VPN, and wish to stop it from running, one can stop the service using the following command:

killall openvpn

This will stop the VPN immediately!

Note Regarding VPNs and Proxychains:

VPNs can be used in conjunction with proxychains, or tor to give oneself an extra layer of privacy, however the tradeoff being that using both in tandem will make one's connection run substantially slower!

How to Disable webrtc:

Overview: **webrtc** can be used to identify one, so it must be disabled. To do so one should open the **firefox browser**, and in their address bar input the following:

about:config

After doing so, one should receive a message that warns them that by proceeding one can potentially void their warranty, of course a hacker should proceed anyway.

Now in the search engine input the following:

media.peerconnection.enabled

Once doing so, **firefox** will bring up said option under **Preference Name**, now to disable **webrtc** under the value section from **true** one needs to change it to **false**, this can be done by clicking on the option.

Changing Ones MAC Address With “macchanger”:

Overview: A **mac address** is used to identify the kind of hardware one is using. When one is on a network, a system administrator will be able to identify your hardware on a network based on the **mac address**.

macchanger is a command in linux that can be used to change, or “spoof” ones **mac address**.

Understanding the mac address format:

A mac address is specific to a **network interface**, or a **network card**; the first three digit sets of a mac address correspond to a specific kind of hardware.

The following is an example of a mac address, and the three dugit sets that identify ones hardware (these digits are in bold):

08:00:27:78:86:27

The above numbers in bold are the three digit sets that identify ones hardware. As soon as one connects to a wireless, or lan network, the mac address can identify them in combination with an IP address within that network.

How to identify ones local mac address:

There are two methods one can use to find out what their current mac address is.

Method One | Utilizing "ifconfig": Within a terminal session, one can use the following command:

ifconfig

This command will bring up ones network interface information, including all their interface devices, ones mac address should be found next to the text **HWaddr**.

Note: The **HWaddr** text is not absolute, it's possible ones mac address could be beside text other than **HWaddr**.

Method Two | Ultilizing "macchanger": Within a terminal session, one can use the following command format:

macchanger -s [network interface device]

Practical Command Format Example:

macchanger -s wlan0

The above command will show ones current eth0 interface mac address, as well as their permanent one.

How to use "macchanger" to set a random mac address:

If one wishes to set a new random mac address, one can do so by inputting the following command into the terminal:

macchanger -r [network interface device]

After one uses the above command a new mac address will be assigned to their network interface device at random.

Note: Upon restarting ones computer, their mac address will revert to the original address.

How to use "macchanger" to list known vendor mac addresses:

To show a list of known vendor mac addresses one may use the following command:

macchanger -l

This list is extremely useful should one wish to change their mac address to a specific vendor address using the **macchanger -m** command option.

How to use “macchanger” to set a custom/specific mac address:

If one would like to set a custom, or specific mac address, they may use the following command format to do so:

macchanger -m [custom mac address] [network interface device]

Using the above command, and inputting a custom mac address after the **-m** option, followed by the desired network interface device will allow one to change the associated device address to sinething specific.

Remember a mac address is **nine digits** long with a **colon** symbol after each **two** (with the exception of the last set). As discussed prior, the first **three digit sets** are what correspond to a specific kind of hardware.

Using the **macchanger -l** command will provide one with a resource of different mac addresses one can use to set a specific mac address. Only the first three digit sets will correspond to specific hardware, the rest of the digits may be inputted at random.

Below is another example if the command to set a custom mac address:

macchanger -m 00:50:08:06:07:05 eth0

How to use “macchanger” to set a random vendor mac address:

If one wishes to set their mac address to a random vendor address, they may do so by using the following command format example:

macchanger -A [network interface device]

The above command will set ones address to a random mac address that will correspond to a vendor.

How to get help for “macchanger”:

For a list of additional macchanger command options, and a helpful overview of the macchanger tool, use the following command:

macchanger -h

Crontab Tutorial + “macchanger” Startup Script:

Overview: The **crontab** command is a way to create a script (referred to as a **cronjob**) to schedule ones computer to perform a specific task.

In this tutorial, the basics of the **crontab** command will be covered, as well as how one can create a script within **crontab** to run **macchanger** on startup to change their mac address each time they start their computer.

Different crontab commands, and explainations:

crontab -e This command allows one to create, or edit crontab scripts.

crontab -l This command will list ones crontab scripts.

crontab -r This command will delete ones crontab scripts.

crontab -r -i This command will ask for verification prior to deleting ones crontab scripts.

crontab -u This command is useful if one wants to use a crontab command on a specific user, for instance:

crontab -u root -e

The above command would allow one to edit the crontab scripts of the user **root**. Utilizing **crontab -u** followed by the user one would like to target, and the command one wants to run for the specified users **crontab**.

Below is another example of how this command may be used:

The above command would list the crontab scripts for the user tyrell.

How to specify a text editor for use with “crontab”:

Different edutors nay be used to edit **crontab** scripts, the following are different commands to set a specific editor to use with the tool.

Specifying nano as the editor for crontab, command:

export VISUAL=nano; crontab -e

Specifying vim as the editor for crontab, command:

export VISUAL=vim; crontab -e

“crontab macchanger” Startup Script Capstone Exercise:

In this capstone exercise, one is going to use **crontab** to create, then run a script that utilizes **macchanger** to change ones mac address on startup.

1). Use the following command to open the crontab script:

crontab -e

2). Once **crontab** is open one should go to the very bottom of the file, and below the following file format **# m h dom mon dow command**, one can then input the following command format example:

@reboot macchanger -A [network interface device]

Practical command format example:

@reboot macchanger -A eth0

Once complete, one should save the file, the above command specifize that on **reboot** (when the computer restarts), the **macchanger -A eth0** command will run; which will change the **eth0** mac address to a random vendor address.

Note: If one wants to add more scripts to the **crontab**, they can simply add them under the last script they've written.

A Deeper Dive Into crontab Structure:

To schedule a command using **crontab** one would use the following syntax format, im the general order they would need to be written:

(m) (h) (dom) (mon) (dow) (command)

The following list will explain each of the format parameters displayed above:

m This field stands for **minute**, and the allowed values range between **0-59**.

h This field stands for **hour**, and the allowed values range between **0-23**.

dom This field stands for **day of the month**, and the allowed values range between **1-31**.

mon This field stands for **month**, and the allowed values range between **1-12**, alternatively one may use names, **mon name ranges** depicted below:

jan, feb, mar, apr, may, jun, jul, aug, sep, oct, nov, dec

dow This field stands for **day of the week**, and the allowed values range between **0-7** (The **0** and **7** values are both used for **Sunday**), alternatively one may use names, **dow name ranges** depicted below:

sun, mon, tue, wed, thu, fri, sat

command Once one has filled out all of their desired scheduling parameters, they can then add the command they would like to run during this time.

Other "crontab" Options:

* Any of the fields can be set to an **asterisk** *, which stands for "first through last". For instance, to run a job every month, put * in the **mon** field.

@reboot Runs ones crontab script once, at startup.

@yearly Runs ones crontab script once a year, same as **0011***.

@annually This option functions the same as "@yearly"

@monthly Runs ones crontab script once a month, same as **001****.

@weekly Runs ones crontab script once a week, same as **00**0**.

@daily Runs ones crontab script once a day, same as **00***.

@midnight This option functions the same as "@daily".

@hourly Runs ones crontab script once an hour, same as **0****.

Important "crontab" Information:

Ranges This stands for the ranges of numbers that are allowed. Ranges are two numbers seperated with a hyphen. The specified range is inclusive; for example **8-12** for an "**hours"** entry specifies execution of hours **8**, **9**, **10** and **11**.

Lists A list is a set of numbers (or ranges) seperated by commas. Examples: **1,2,5,9,1-4,8-12**

Note: This is useful if one wants to run a command at a specified time, then again later at another specified time; example below:

09, 18 * * mon /home/pron/genkigenki.sh

The comma makes it so the command will run at 9:00am, and 6:00pm, every Monday.

Step values Step values can be used in conjunction with ranges. For example, **0-23/2** in the **hours** field means “every other hour”. Steps are also permitted after an asterisk, if one wants to specify “every two hours”, they may use ***/2**.

Note: The day of a commands execution can be specified by two fields: **day of the month**, and **day of the week**. If both fields are restricted (in other words, they aren’t *), the command will be run when rither field matches the current time, for example:

30 4 1,15 * 5

Would cause the command to be executed at 4:30am on the 1’st, and 15’th of each month, plus every Friday.

Crontab examples:

00 09-18 * /home/carl/hourly-archive.sh**

The above would run **/home/carl/hourly-archive.sh** every hour, on the hour, from 9:00am (09:00) through 6:00pm (18:00), every day.

30 22 ** mon,tue,wed,thu,fri /usr/local/bin/backup

The above would run **/usr/local/bin/backup** at 10:30pm, every weekday.

Other Opsec Tools:

Overview: This section will cover various tools useful to operate ones linux system securely, it's far from comprehensive, but provides information on basic tooling I feel is beneficial from a defensive, and system hardening standpoint.

Understanding, and verifying hashes: When downloading a file, or (uploading a file) one can obtain the files associated hash, a hash can best be described as a file signature, and is used to validate that the associated program one downloads has not been modified.

This hash should then be compared to the hash listed publically from the files source provider (usually listed on the file providers website).

This relies on the hash that is being provided to come from a trusted source, and vice versa, when one provides the hash of a file they upload, they likewise should to be trusted by those downloading the file.

Note: If the hash one obtains, differs from the publically listed hash from the trusted file provider, it is not safe to install, and should be deleted.

To obtain a files hash within a terminal session, one can run the following commands depending on what kind of hash one wishes to retrieve:

Retrieving a file md5 hash command format example:

md5sum [file]

Retrieving a file SHA-1 hash command format example:

sha1sum [file]

Retrieving a file SHA-256 hash command format example:

sha256sum [file]

Retrieving a file SHA-512 hash command format example:

sha512sum [file]

Running file reputation checks against hashes:

There are free file reputation services available online one can leverage to determine whether a file is dangerous, these services work be checking the provided has against a known list of other hashes that belong to malicious programs, one of the better known services is **talos**, the website for this service is listed below:

https://www.talosintelligence.com/talos_file_reputation

Using "top" to observe system resource use: The **top** tool is used to observe linux system resources, and is brst leverage to optimize system performance, on investigate what programs are heavily ulitizing a system resources. This is one way to try to determine if your computer has been compromised by malicious software, or to alternative perform forensics on a VM sandboxed malware.

The **top** tool has many options that can provide more insight into running processes, these should be explored, but to run top for a basic overview of a systems resource usage, one simply needs to use the following command:

top

Checking for rootkits using "rkhunter": The **rkhunter** tool, is an open source program that can be leveraged on linux systems to search for common malicious files, and as the name suggests rootkits.

Installing "rkhunter" tool command:

sudo apt install rkhunter

“rkhunter” tool full scan command:

sudo rkhunter –checkall

Note: The output returned from the **rkhunter** tool should be investigated, but it should not be considered definitive, as **false positives** are known to be reported in the scan output.

Generating a strong password using “pwgen”: The password gen tool can be used to quickly create a strong random password, this password will be extremely difficult for an attacker to crack. The downside is one is going to need to remember this complex password, but for OPSEC purposes this tool is fantastic for secure password development, just don't forget the password:

Installing “pwgen” tool command:

sudo apt install pwgen

Generating a secure random password with “pwgen”:

pwgen -s 20 1

Conclusion:

This concludes this guide, the next guide will build on this one, and explain the fundamentals of WiFi penetration testing tools used to attack WEP, WPA2, and WPS implementations, as well as how to secure wifi networks. I may add screenshots, and additional content at a later date, if I do I'll add a revision section near the start of this document.

Two big things I thought about including in this guide, but feel would be better suited to there own walkthrough are the development of a partitioned Kali live LUKS/LVM instance, with persistence, and a nuclear option. And an explanation on IP tables, and how to build a firewall, but I feel it’s out of the scope for this current skill level.

Thank you for reading, well done on making it to the end! Please follow, and like this post if you want to get updates as I release more guides, and infosec related content.